UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.	Case No. CV 10-03561 WHA
Plaintiff,	
v.	
GOOGLE INC.	
Defendant.	

SUMMARY AND REPORT OF ROBERT ("BOB") G. VANDETTE

**SUBMITTED ON BEHALF OF PLAINTIFF
ORACLE AMERICA, INC.**

IV.   PERFORMANCE ANALYSIS OF THE '104 PATENT ON ANDROID

A.   Google Android benefits from RE38,104

26.   I ran experiments to disable the Android functionality that Oracle accuses of infringing the '104 patent.

27.   The baseline for these experiments was the Froyo release of Android, pulled from the Google git repository with the Google repo commands

```
$ repo init -u git://android.git.kernel.org/platform/manifest.git -b froyo
$ repo sync
```

to initialize and sync from the Google repository.

28.   I then made two additional copies of the repository

(1) building side tables of resolved constant pool entries, but not quickening instructions

(2) not building side tables of resolved constant pool entries, nor quickening instructions

29.   Modifications were made to the source in the copies to implement the experiments.

30.   The modification for the first experiment (side tables but no quickening) are restricted to

dalvik/vm/analysis/DexOptimize.c

and consist of bracketing with #ifdef and #endif code that rewrites instructions to their QUICK forms. The modifications for the first experiment are shown in '104 Appendix 1.

31.   The modifications for the second experiment (neither side tables nor quickening) included all the changes for the first experiment, plus additional changes to

dalvik/vm/analysis/DexOptimize.c

and some changes to

dalvik/vm/oo/Resolve.c

8

32.   The changes for the second experiment also consist of bracketing with #ifdef and #endif code that builds the side tables of resolved constant pool entries. The modifications for the second experiment are shown in '104 Appendix 2.

33.   Each workspace was compiled with

```
$ make -j2
```

34.   The benchmarks used to test the performance of the workspaces were

(1) CaffeineMark™ 3.0 * (http://www.benchmarkhq.ru/cm30)

(2) SciMark 2.0 (http://math.nist.gov/scimark2)

(3) kBench (Sun/Oracle Internal Benchmark)

35.   All the benchmarks were converted from standard jar files to dex jar files with, for example

```
$ dx --dex --verbose --output=scimark-dex.jar scimark.jar
```

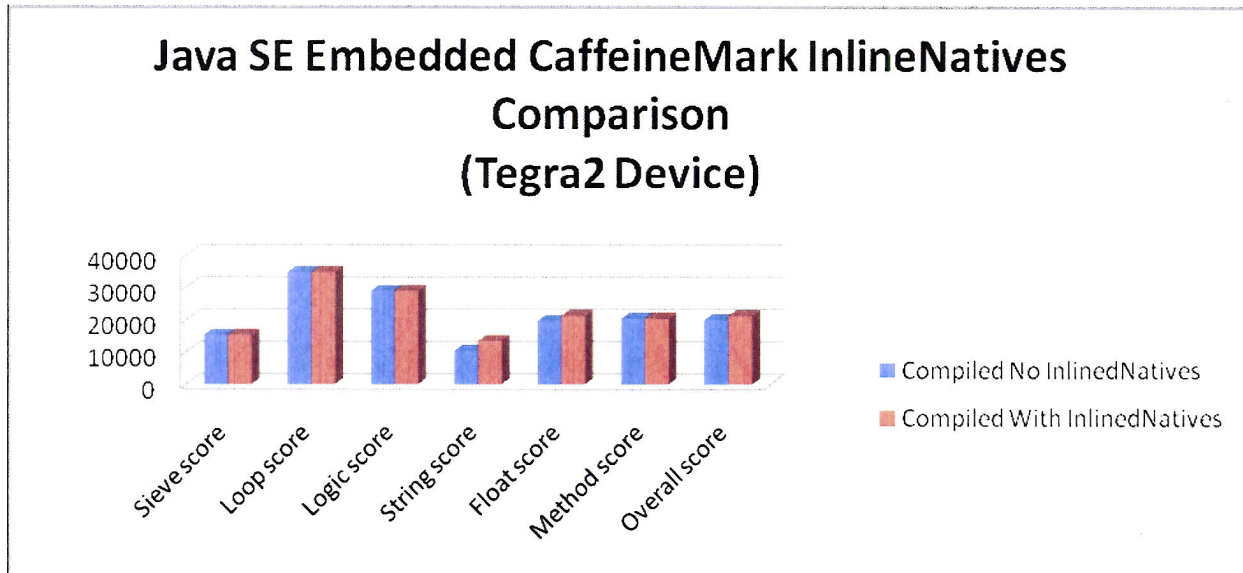36.   I tested the performance of the repositories by running on the supplied Android emulator with

```
# dalvikvm -Xint:fast -cp /data/app/cm3-dex.jar CaffeineMarkEmbeddedApp
# dalvikvm -Xint:fast -cp /data/app/scimark-dex.jar scimark 2.0
# dalvikvm -Xint:fast -cp /data/app/kBench-dex.jar RunAll
```

I did not try running the trace compiler, because an examination of the source code indicated that it would not run without the side tables of resolved constant pool entries.

37.   Each of the three workspaces ran all three of the benchmarks. I ran 10 iterations of each benchmark.

38.   Separately, I built new workspaces with the modified source files, and ran them on a Beagleboard.

39.   The accompanying charts attached as Exhibit E record the results of runs with the -Xint:fast command line option on the Beagleboard.

9

Java SE Embedded CaffeineMark InlineNatives Comparison (Tegra2 Device)

65.   The chart above shows a 1.15X improvement in the String sub test.

C.   Conclusion

66.   The data presented in this report shows that Java SE benefits from the application of the '205 patent by 13X when the JIT compiler is enabled in Java SE Embedded and by 1.15X for the String sub test when inlined natives are enabled.

I declare under penalty of perjury that the foregoing is true and correct.

Dated: August 8, 2011

Robert Vandette